SECTION 2 SUMMARY OF REQUIREMENTS

2.1 SYSTEM DESCRIPTION.

- **2.1.1 Existing Procedures.** Reuse not only applies to source code, but also to other products of the development life-cycle, which may include: requirements documentation, system specifications, user manuals, design documentation, test suites, and any information to support development of software. Currently, asset reuse is limited to an individual's knowledge of ad hoc, localized methodologies that are customarily project specific, thereby discouraging the transfer of technology from one project or department to another.
- **2.1.2 System Requirements.** The system will provide a repository in which multiple users can simultaneously search for, evaluate, and acquire reusable assets and in which the DSRS support staff can perform various maintenance activities for the repository of assets.

The DSRS will provide a structured querying mechanism with a user-friendly interface. Figure 2-1 displays the basic DSRS architecture, which lists each executable and the mapping of each executable to the user interface, DSRS software, Database Management System (DBMS), and the operating system.

The User Tool will be implemented on a Sun/UNIX workstation to support a Motif-based interface; implemented for a PC/Windows workstation to support a Microsoft Windows-based interface; and operational on a PC/Windows NT interface. The User Tool functionality will be consistent between each interface supported, unless limited by the host environment.

The Librarian Tool will be implemented for a PC/Windows workstation to support a Microsoft Windows-based and a Windows-NT interface.

The DSRS Server will be built on a relational database, which organizes information about the reusable assets. The reusable asset files (including the actual asset, an abstract file, and document files) will be stored in the underlying file system provided by the operating system software and will be transparent to the user.

DSRS Applications	User		Librarian
Client Executables	DSRS	DSRS	DSRSLIB
Client HW/OS*	PC/DOS	Sun/UNIX	PC/DOS
User Interface	MS-Windows/ Windows-NT	X/Motif	MS-Windows/ Windows-NT
DBMS	Oracle/Access	Oracle/MSQL	Oracle
Server Executables	pc_server or pc_server.sol23 and dsrs_server or dsrs_server.sol23		lib_server or lib_server.sol23
Server HW/OS*	Sun/UNIX		

^{*} Refer to Section 5.2 for environment specifics.

Figure 2-1. Basic DSRS Architecture

2.1.2.1 <u>User Tool.</u> Figure 2-2 shows the User Tool process. The following paragraphs briefly discuss, at a high level, each of the major functions.

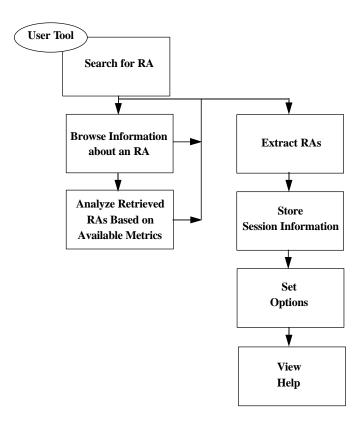


Figure 2-2. User Tool Process

- **2.1.2.1.1** <u>Search Mechanisms</u>. The user's main interaction with the system will be searching for an RA. The user, having determined in advance the objective of the search, will use the following types of search mechanisms to help build a Candidate List of RAs:
 - a. **List RA IDs.** The system will create a list of candidate RAs based on the domain(s) selected by the user and the RA ID or wildcard RA ID entered by the user.
 - b. **List RA Names.** The system will create a list of candidate RAs based on the domain(s) selected by the user and the RA Name or wildcard RA Name entered by the user.
 - c. **Keyword Search.** The system will create a list of candidate RAs based on the Domain(s) selected and the keyword(s) entered by the user. The system will search the repository for RAs in which the keywords can be found in the key file.
 - d. **Catalog Indexes (WWW).** The system will display the indexed catalog of RAs via a World Wide Web (WWW) browser previously specified by the user.
- **2.1.2.1.2 Browse RA Information.** The user may choose to browse information about a candidate RA. Such information will include the RA file list (including abstract, source, document, or problem report files), metrics, and related RAs. The user will also be able to view information stored on a WWW page about the selected RA.
- **2.1.2.1.3** Tools. The system will provide the capability to display a list of metrics, analyze the candidate RAs based on a metric, clear the local catalog information, and sort the Candidate List.
- **2.1.2.1.4** Analyze. The user may also choose to analyze the list of retrieved RAs based on available metrics and sort them according to the available options. The sort may be in ascending or descending order or none (as it appears in the candidate list).
- **2.1.2.1.5** <u>Clear Local Catalog.</u> The user may choose to clear the information stored in the local database.
- **2.1.2.1.6** Sort List. The user may choose to sort the list of candidate RAs based on the RA ID or RA Name.
- **2.1.2.1.7** Extract RAs. The user may select multiple RAs from the Candidate List to be extracted. Once the user selects the RAs, the system will identify any RAs related to each RA selected. Depending on the user's access rights, the system will also allow extraction of the related RAs. Users may extract assets interactively from the local site, remote DSRS sites, or foreign sites (other interoperable repositories) via the Transfer Control Protocol/Internet Protocol (TCP/IP) Asset Transfer Interface. The extraction process is illustrated in Figure 2-3.

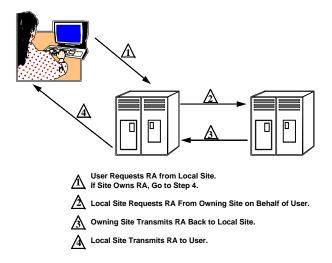


Figure 2-3. Extract Process

Users' access to assets will be determined by the assignment of group identifiers. Asset access will be managed by the Librarian assigning group identifiers to site IDs, user IDs, and RA IDs. A group ID will be a tag identifying an abstract collection of sites, users, and assets and conveying certain rights and privileges to the members of that collection. Asset access for groups is depicted in Figure 2-4.

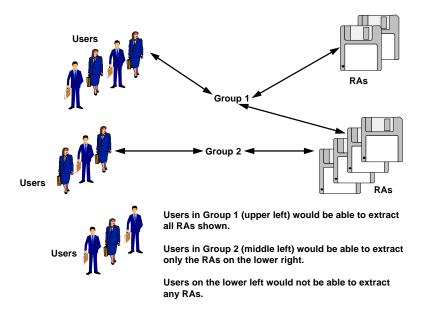


Figure 2-4. Asset Access for Groups

- **2.1.2.1.8** <u>Sessions</u>. The user will be able to retrieve and save session information containing the RA ID and RA Names in the Candidate List and the domains used to create the Candidate List.
- **2.1.2.1.9 Options.** The system will allow users to set options for the following:
 - a. **Change Password**. The system will allow users to change their password.
 - b. **Preferences.** The system will allow users to select a default domain (used when searching), set the threshold notification, and select the default text browser (used when viewing text RA files).
 - c. **Network.** The system will allow users to enter network configuration information to be used when connecting to the DSRS server. The system will also allow users to test their network communication to the server.
 - d. **WWW.** The system will allow users to enter configuration information for the WWW browser to be used with functions that launch a WWW browser.
- **2.1.2.1.10** <u>Help.</u> The system will provide on-line help and a tutorial for beginner users. The system will provide the user with the capability to search the on-line help by selecting from a list of available topics.
- **2.1.2.2** <u>Librarian Tool</u>. The Supervisor/Librarian for the DSRS will be responsible for maintaining the assets in the repository and the information that supports the operation of the system. Figure 2-5 shows the Librarian Tool process. The following paragraphs briefly discuss, at a high level, each of the major functions.
- **2.1.2.2.1 RAs.** The Supervisor/Librarian will be able to maintain the RAs and RA information in the local catalog. The RA information will include RA to Domain Assignments, RA Metrics, relationships with other RAs, and RA Files.
- **2.1.2.2.2** Groups. The Supervisor/Librarian will be able to maintain group information and group assignments for the local, remote and foreign sites. Groups will be used to control access to assets of the local repository to local users by designating users, sites, and RAs collectively. Groups will be used to control access to assets of the local repository to remote and foreign site users by designating sites. The remote or foreign site will designate which users will be authorized to access the group of assets.
- **2.1.2.2.3** <u>Users.</u> The Supervisor/Librarian will be able to maintain user information and assignments for local users. User IDs and passwords will be used to validate and control access to the system.
- **2.1.2.2.4** Sites. The Supervisor/Librarian will be able to maintain the information for local and remote DSRS sites, and for foreign sites (other interoperable repositories). Site IDs are used to help formulate and validate RA IDs and will be used to validate sites to determine access rights to RAs.

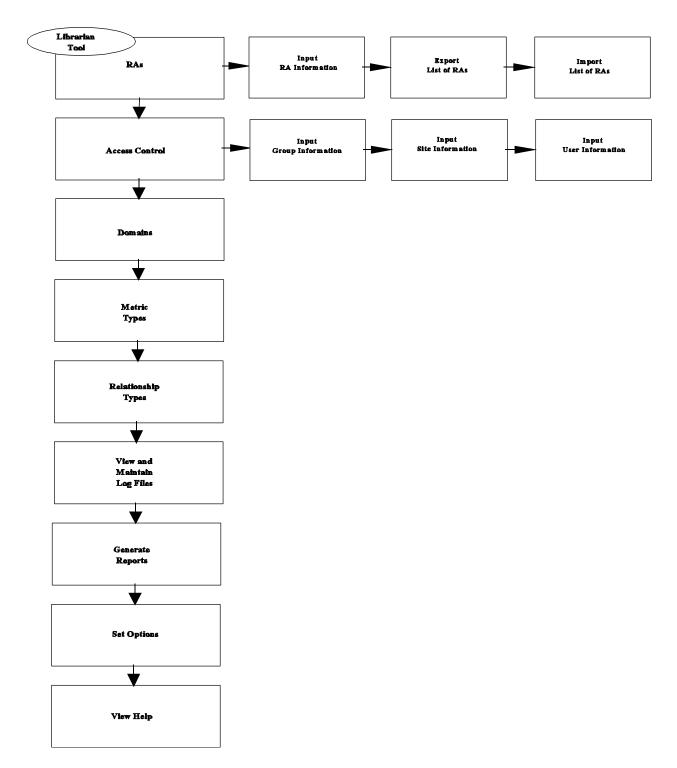


Figure 2-5. Librarian Tool Process

- **2.1.2.2.5** <u>Domains</u>. The Supervisor/Librarian will be able to maintain the domains used for the classification or assignment of RAs.
- **2.1.2.2.6** <u>Metrics.</u> The Supervisor/Librarian will be able to maintain the types of metrics indigenous to the local site.

- **2.1.2.2.7 Relationships.** The Supervisor/Librarian will be able to maintain the types of relationships that can be used when entering related RAs.
- **2.1.2.2.8** <u>Logs.</u> The Supervisor/Librarian will be able to maintain the entries in the Usage Log which will record user extraction of RA files and requests for tapes and hard copies of RAs. Supervisor requests for RAs will not be recorded in the Usage Log.
- **2.1.2.2.9** <u>Import.</u> The Supervisor/Librarian will be able to read RA information contained in an export file (created at another DSRS) and write the information to the local catalog (the local database). The Import function will verify that the file is in the correct DSRS export file format.
- **2.1.2.2.10** Export. The Supervisor/Librarian will be able to submit assets to other DSRS sites via the Export function. The Export function will create an export file containing RAs and associated information to be imported into another DSRS catalog. The Export function will also create index files to be imported into other interoperable repositories (foreign sites).
- **2.1.2.2.11 Reports.** The Supervisor/Librarian will have the capability to generate predefined or customized reports on the data stored in the database.
- **2.1.2.2.12 Options.** The Supervisor/Librarian will have the capability to enter configuration information for the server's host name and database.
- **2.1.2.2.13** <u>Help.</u> The system will provide on-line help and a tutorial for beginner users. The system will provide the user with the capability to search the on-line help by selecting from a list of available topics.
- **2.1.2.3 Server.** The Server will process the following:
 - a. **Authentication Requests.** The server will authenticate users logging in from the DSRS User Tool and the DSRS Librarian Tool.
 - b. **Catalog Requests.** The server will process catalog update requests from the DSRS User Tool and the DSRS Librarian Tool. These requests will be used to facilitate consistency between the server database and client cache database.
 - c. **File Transfer Requests.** The server will process file transfer requests from the DSRS User Tool, remote DSRS sites, and foreign sites (other interoperable repositories). Figure 2-3 illustrates the processing that occurs when files are transferred between servers and a local client.

- **2.2 SYSTEM FUNCTION.** This section describes the Top-Level Software Units (TLSUs) and the interrelationships among those TLSUs used in the implementation of the DSRS. According to the Booch method, a TLSU is represented by a class.
- 2.2.1 <u>Classes (TLSUs)</u>. A class captures the common structure and common behavior of a set of objects. It is an abstraction of real-world items. Each class encapsulates the subset of operations necessary to implement the system. Refer to Section 4.3.5 to view the complete list of operations for each class. This section describes all classes comprising the DSRS.
- 2.2.1.1 <u>User Tool</u>. The *User Tool* is implemented on three platforms: MS-Windows, Windows-NT, and X/Motif. However, there are three executables produced: User Tool for MS-Windows which also operates on Windows-NT platforms, User Tool for Unix SunOS, and User Tool for Unix Solaris. The Unix SunOS version and Unix Solaris version are identical in their design and implementation. Therefore, those two versions are covered under the heading: X/Motif. The following paragraphs describe the classes under their appropriate headings.
- $2.2.1.1.1 \, \underline{\text{MS-Windows and Windows NT}}.$ The top-level classes for the DSRS User for MS-Windows and Windows-NT are as follows:
- 2.2.1.1.1.1 <u>DSRSUserApp</u>. *DSRSUserApp* is a container class. It functions as a folder containing all classes and other files related to the application. There are no operations associated with it.
- 2.2.1.1.1.2 <u>FrmBrMet</u>: FrmBrMet is a class representing the Browse Metrics screen. Its function is to enable the user to browse the metric(s) of an RA.
- 2.2.1.1.1.3 <u>FrmWldName</u>. *FrmWldName* is a class representing the *List RA Names* screen. Its function is to allow the user to search for RAs by RA Name and list the results.
- 2.2.1.1.1.4 <u>FrmWldID</u>. *FrmWldID* is a class representing the *List RA IDs* screen. Its function is to allow the user to search for RAs by RA ID and list the results.
- 2.2.1.1.1.5 <u>FrmNetDialog</u>. *FrmNetDialog* is a class representing the *Network Configuration* screen. Its function is to allow the user to change the default network configuration.
- 2.2.1.1.1.6 <u>FrmLogin</u> is a class representing the *Login* screen. Its function is to enable the user to connect to a server.
- 2.2.1.1.1.7 <u>FrmAbout</u>. *FrmAbout* is a class representing the *About* screen. It contains information about the application, such as version number and release date.
- 2.2.1.1.1.8 <u>FrmExtract</u>: FrmExtract is a class representing the Extract RAs screen. Its function is to allow the user to select multiple RAs from the candidate list to be extracted as described in Figure 2-3.

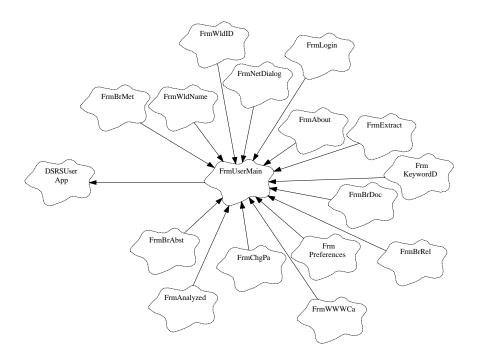
- 2.2.1.1.1.9 <u>FrmKeywordD</u> is a class representing the *Keyword Search* screen. Its function is to allow the user to search for RAs based on keywords entered, and list the results.
- 2.2.1.1.1.10 <u>FrmBrDoc</u>. *FrmBrDoc* is a class representing the *Browse File List* screen. Its function is to enable the user to browse the list of files available for an RA and view the content.
- 2.2.1.1.1.11 <u>FrmBrRel</u>. *FrmBrRel* is a class representing the *Browse Related RA* screen. Its function is to enable the user to browse the list of related RA(s) and the kind of relationship(s) for a selected RA.
- 2.2.1.1.1.12 <u>FrmPreferences</u>. *FrmPreferences* is a class representing the *Preferences* screen. Its function is to allow the user to change the default domain, search threshold notification, and default text viewer.
- 2.2.1.1.1.13 <u>FrmWWWCa</u> is a class representing the *World Wide Web Catalog* screen. Its function is to enable the user to select the World Wide Web browser and determine the default home page.
- 2.2.1.1.1.14 <u>FrmChgPa</u>. *FrmChgPa* is a class representing the *Change Password* screen. Its function is to allow the user to change the password.
- 2.2.1.1.1.15 <u>FrmAnalyzed</u>. *FrmAnalyzed* is a class representing the *Analyze* screen. Its function is to allow the user to analyze retrieved RAs based on available metrics and sort them according to the available options.
- 2.2.1.1.1.16 <u>FrmBrAbst</u>. *FrmBrAbst* is a class representing the *Abstract* screen. Its function is to allow the user to view the abstract of an RA.
- 2.2.1.1.1.17 <u>FrmUserMain</u>. *FrmUserMain* is a class representing the main screen of the application. Its function is to allow the user to access the functionalities of the *User Tool* described in Figure 2–2 through menu items of the menubar or command buttons on the toolbar.
- 2.2.1.1.2 X/Motif. The top-level classes for the DSRS User for X/Motif are as follows:
- 2.2.1.1.2.1 <u>UDSRSApp</u>. *UDSRSApp* is a class representing the application—level class, which is the highest in the application framework hierarchy. It is an abstract class and cannot be instantiated. Each application can have one application object only. This class handles the start—up and shut—down activities of an application.
- 2.2.1.1.2.2 <u>USession</u>. *USession* is a class representing the document-level class. It resides at the second layer of the application framework hierarchy. This class functions as a central means of communication for changes and updates in different window-level classes.

- 2.2.1.1.2.3 <u>USessWin</u>. *USessWin* is a window-level class representing the main screen of the application. Its function is to allow the user to access the functionalities of the *User Tool* described in Figure 2-2 through menu items of the menubar or command buttons on the toolbar.
- 2.2.1.1.2.4 <u>UDSRSWin</u> is a window-level class representing the *About* screen. It contains information about the application, such as version number and release date.
- 2.2.1.1.2.5 <u>UWidWin</u> is a window-level class representing the *List RA IDs* screen. Its function is to allow the user to search for RAs by RA Names and list the results.
- 2.2.1.1.2.6 <u>UPrefWin</u> window-level class representing the *Preferences* screen. Its function is to allow the user to change the default domain, search threshold notification, and default text viewer.
- 2.2.1.1.2.7 <u>UCngPWin</u> is a window-level class representing the *Change Password* screen. Its function is to allow the user to change the password.
- 2.2.1.1.2.8 <u>UWWWWin</u>. *UWWWWin* is a window-level class representing the *World Wide Web Catalog* screen. Its function is to enable the user to select the World Wide Web browser and determine the default home page.
- 2.2.1.1.2.9 <u>URelWin</u>. *URelWin* is a window-level class representing the *Browse Related RA* screen. Its function is to enable the user to browse the list of related RA(s) and the kind of relationship(s) for a selected RA.
- 2.2.1.1.2.10 <u>UMetricWin</u> is a window-level class representing the *Browse Metrics* screen. Its function is to enable the user to browse the metric(s) of an RA.
- 2.2.1.1.2.11 <u>UAbstWin</u>: *UAbstWin* is a window-level class representing the *Abstract* screen. Its function is to allow the user to view the abstract of an RA.
- 2.2.1.1.2.12 <u>GNetWin</u> is a window-level class representing the *Analyze* screen. Its function is to allow the user to analyze retrieved RAs based on available metrics and sort them according to the available options.
- 2.2.1.1.2.13 <u>UXtractWin</u>. *UXtractWin* is a window-level class representing the *Extract RAs* screen. Its function is to allow the user to select multiple RAs from the candidate list to be extracted as described in Figure 2-3.
- 2.2.1.1.2.14 <u>UFileWin</u>. *UFileWin* is a window-level class representing the *Browse File List* screen. Its function is to enable the user to browse the list of files available for an RA and view the content.

- 2.2.1.1.2.15 <u>UFindWin</u>: *UFindWin* is a window-level class representing the *Keyword Search* screen. Its function is to allow the user to search for RAs based on keywords entered, and list the results.
- 2.2.1.1.2.16 <u>UNamesWin</u>. *UNamesWin* is a window-level class representing the *List RA Names* screen. Its function is to allow the user to search for RAs by RA ID and list the results.
- 2.2.1.1.2.17 <u>ULogWin</u>. *ULogWin* is a window-level class representing the *Login* screen. Its function is to enable the user to be connected to a server.
- 2.2.1.1.2.18 <u>UNetWin</u> is a window-level class representing the *Network Configuration* screen. Its function is to allow the user to change the default network configuration.
- 2.2.1.1.2.19 <u>UServer</u>: a control class which handles the communication between the DSRS User for X/Motif and the server code.
- 2.2.1.1.2.20 <u>ULocalDB</u>. *ULocalDB* is a control class which handles the communication between the DSRS User for X/Motif and the local database (MSQL).
- 2.2.1.2 <u>Librarian Tool</u>. The *Librarian Tool* is implemented in one platform only: MS-Windows. The following paragraph lists and describes top-level classes developed for the DSRS Librarian for MS-Windows.
- 2.2.1.2.1 <u>DSRSLibApp</u> is a container class. It functions as a folder containing all classes and other files related to the application. There are no operations associated with it.
- 2.2.1.2.2 <u>FrmLibMain</u>. *FrmLibMain* is a class representing the main screen of the application. Its function is to allow the Librarian/Supervisor to access the functionalities of the *Librarian Tool* described in Figure 2–5 through menu items of the menubar or command buttons on the toolbar.
- 2.2.1.2.3 <u>FrmLogin</u> is a class representing the *Login* screen of the application. Its function is to enable the Librarian/Supervisor to connect to a server and a database.
- 2.2.1.2.4 <u>FrmRA</u> is a class representing the <u>RAs</u> screen of the application. Its function is to enable the Librarian/Supervisor to maintain the RAs and RA Information in the local catalog.
- 2.2.1.2.5 <u>FrmGroup</u> is a class representing the *Groups* screen of the application. Its function is to enable the Librarian/Supervisor to maintain group information and group assignments for RAs, users, and sites.
- 2.2.1.2.6 <u>FrmUsers</u>. *FrmUsers* is a class representing the *Users* screen of the application. Its function is to enable the Librarian/Supervisor to maintain user information and group assignments for local users.

- 2.2.1.2.7 <u>FrmSites</u>. *FrmSites* is a class representing the *Sites* screen of the application. Its function is to enable the Librarian/Supervisor to maintain the information for local, remote DSRS sites, and foreign sites (other interoperable repositories).
- 2.2.1.2.8 <u>FrmNewDomain</u>. *FrmNewDomain* is a class representing the *Domains* screen of the application. Its function is to enable the Librarian/Supervisor to maintain the domains.
- 2.2.1.2.9 <u>FrmMetrics</u> is a class representing the <u>Metrics</u> screen of the application. Its function is to enable the Librarian/Supervisor to maintain the types of metrics for the local site.
- 2.2.1.2.10 <u>FrmRelation</u>. *FrmRelation* is a class representing the *Relationships* screen of the application. Its function is to enable the Librarian/Supervisor to maintain the types of relationships which can be used when entering related RAs.
- 2.2.1.2.11 <u>FrmImport</u>. *FrmImport* is a class representing the *Import* screen of the application. Its function is to enable the Librarian/Supervisor to read RA information contained in an export file (created at another DSRS) and write the information to the local catalog.
- 2.2.1.2.12 <u>FrmExport</u>: FrmExport is a class representing the Export screen of the application. Its function is to enable the Librarian/Supervisor to submit assets to other DSRS sites via the Export function. The export function will create an export file containing RAs and associated information that will be copied (imported) into another DSRS catalog. The Export function will also create indexed files that will be copied (imported) into other interoperable repositories (foreign sites).
- 2.2.1.2.13 <u>FrmUsage</u>. *FrmUsage* is a class representing the *Usage* screen of the application. It functions as a log of RA extractions.
- 2.2.1.2.14 <u>FrmAbout</u> is a class representing the *About* screen of the application. It contains information about the application, such as version number and release date.
- 2.2.2 <u>DSRS Class Diagram with Relationships</u>. Classes do not exist in isolation. Rather they are related in a variety of ways to form the class structure of the domain. Relationships help further define the classes by exposing their content or dependency on the content of others. Figure 2-6, Figure 2-7, and Figure 2-8 illustrate class diagrams with relationships. These class diagrams are the actual diagrams used in implementation. They look different from the class diagrams produced during the analysis stage. Refer to Section 4 for detailed information about the analysis stage.

Figure 2-6. DSRS User for MS-Windows Class Diagram with Relationships



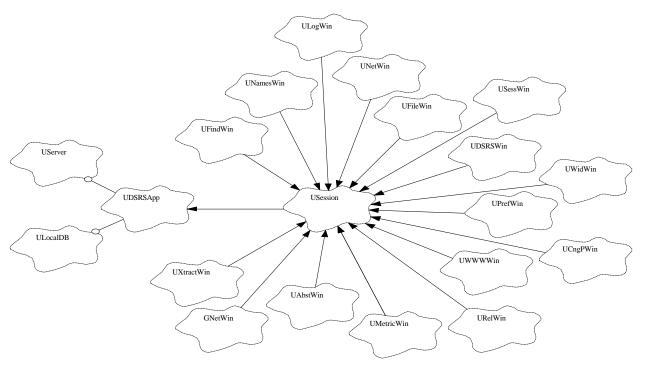


Figure 2-7. DSRS User for X/Motif Class Diagram with Relationships

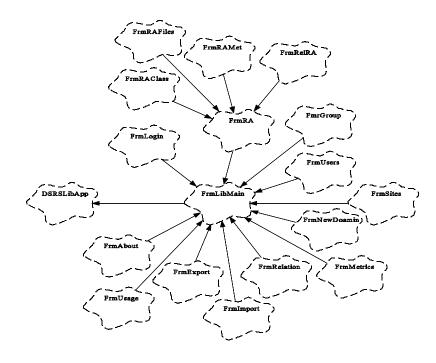


Figure 2-8. DSRS Librarian for MS-Windows Class Diagram with Relationships